

## Узкие места и потенциальные проблемы масштабирования приложения и способы их решения:

### 1) Обработка реального времени в *Game Service*:

- а) Высокая нагрузка на CPU;
- б) Latency при передаче команд;
- в) Синхронизация между экземплярами.

Решение - использовать горизонтальное масштабирование с шардированием по gameId, а также выделять отдельные инстансы для разных типов боев;

### 2) Пиковые нагрузки в *Tournament Service*:

- а) Высокая нагрузка на БД во время массовой регистрации;
- б) Сложности формирования турнирной сетки при большом и интенсивно динамически увеличивающемся количестве участников;
- в) Возможный Race Condition при ограниченности ресурсов турнира.

Решение - использовать асинхронную обработку заявок из очереди, кэширование турнирных сеток.

### 3) Матчмейкинг в *Tournament Service*:

- а) Длительное время подбора соперников;
- б) Неоптимальные пары игроков;
- в) Сложность балансировки команд.

Решение - постоянное совершенствование алгоритма матчмейкинга.

### 4) Массовые уведомления в *Notification Service*:

- а) Очереди сообщений;
- б) Потеря уведомлений;
- в) Задержки доставки.

Решение – приоритетные очереди для разных типов уведомлений, batch processing для массовых рассылок.

### 5) Согласованность данных между сервисами *GameService*, *RatingService* и *TournamentService*:

- а) Несвоевременное обновление рейтингов;
- б) Расхождение в статистике игроков;
- в) Дублирование обработки результатов.

Решение – паттерн Saga для распределенных транзакций.

### 6) Большой объем данных по уже прошедшим турнирам в *ReplayService*:

- а) Высокая нагрузка на хранилище.

Решение - компрессия данных, шардирование по данным и турнирам.